Homework assignment #3 (452 &352),

due on 4/20/2022, Wednesday

In case you meet difficulties, there are hints provided at the end of this file.

Q1. (10 points)

The code below using basic loop. Please edit the codes, use cursor for loop (“For your\_index IN cursor\_name LOOP”). The output of your new codes should be the same as the original program.

DECLARE

CURSOR c IS

SELECT last\_name ||', ' || first\_name Full\_name, salary,

department\_ID

FROM employees

WHERE salary < 2500

ORDER BY salary DESC;

Emp\_Name varchar2 (46);

sal employees.salary%TYPE;

deptid employees.department\_id%TYPE;

BEGIN

DBMS\_OUTPUT.put\_line

(' No Emp Full Name Salary Dept ID');

DBMS\_OUTPUT.put\_line

('---- ---------------------- -------- ---------');

OPEN c;

LOOP

FETCH c INTO Emp\_Name, sal, deptid ;

EXIT WHEN c%NOTFOUND;

DBMS\_OUTPUT.put\_line (rpad ( c%ROWCOUNT, 7) ||

RPAD (Emp\_Name, 25) || to\_char (sal, '$99,999')

||' '|| deptid);

END LOOP;

CLOSE c;

END;

Q2. (25 points)

Assume that the company has decided a one-time bonus for the employees in department 80 only, the amount is decided as below.

For employees that get commission\_pct equal to or great than .25:

if salary > = 10000 then bonus := 1000

if salary >= 7000 and salary < 10000 then bonus := 800

If salary < 7000 then bonus := 600

For employees that get commission\_pct less than .25 and equal and great than .15:

if salary > = 10000 then bonus := 700

if salary >= 700 and salary < 10000 then bonus := 600

If salary < 7000 then bonus := 500

All other employees in department 80 will get $450.

Write an anonymous PL/SQL program, retrieve the info needed for each employee (department 80 only), calculate the amount of bonus he/she should get. Your program will accumulate the bonus from each employee, print out the total amount of the bonuses. (for submission, no need to print out individual bonus)

Q3. (25 points)

This question will use some view(table) in the system catalog: “All\_tab\_columns”.

“All\_tables” lists all tables that user can read, but it does not list the details of the columns in each table. In the view “All\_tab\_columns”, it lists the “owner” of the tables, the “table\_name” and information of columns under each table, such as the “column\_name”, the column “data\_type” and the length of data\_type of that column. The command “desc All\_tab\_columns” will display the info related. Do not simply “select \* from all\_tab\_columns”, the output is not easy to read. You need to use SQL Plus commands to format the output, such as:

column Owner format A10

column table\_name format A30

In this question, you will write an anonymous PL/SQL program. The program will define a cursor with two parameters. The select statement will retrieve the column\_name, data\_type and data\_length from the view All\_tab\_columns for a certain “owner” and a certain “table\_name” that will be passed to it through the parameters.

In the executable section, you will open that cursor two times, first time with the parameters of USER and ‘EMPLOYEES’, the second time with USER and ‘DEPARTMENTS’. Your program will print out the information you have retrieved.

Following lines are just for your reference, not needed for this question. These SQL and SQL Plus commands will display the table-column info of all tables you have created:

Column table\_name format A25

Column column\_name format A25

Column data\_type format A15

select table\_name, column\_name, data\_type, data\_length

from all\_tab\_columns

where owner = user;

Here “user” is the current login account.

Q4. (20 points) Cursor for update, based on table employees.

Write an anonymous PL/SQL program, it will increase the salary 11% of their current salary (it means salary will be salary \* 1.11 ) for those employees whose salary less than 2,450 and without commission.

Your program should declare a cursor with “FOR UPDATE” request, thus it can request the system to lock those records retrieved and change these records later.

After update, print out some info about those affected employees, display their ID, last name, old salaries (before this time increase) and new salary.

It is better to rollback after the print out and before the end of program. (in the real life, program should commit the changes if everything runs fine).

Q5. (20 points) Cursor Variable

Define a strong cursor variable type, name it EmpCurTyp. Its return type is

employees%ROWTYPE, based on table Employees.

Then declare a cursor variable of this type.

In your program, first you will open the cursor variable for a select statement as

SELECT \* FROM employees

WHERE department\_id = deptid

ORDER BY last\_name;

Where deptid is a variable of data type as number with an initial value 30.

print out the employees’ employee ID and full names in this department 30.

(please pay attention, the Cursor-For loop does NOT work for cursor variable)

Then, open that cursor variable again, this time for the statement of

SELECT \* FROM employees

WHERE commission\_pct is null and salary > 15000

ORDER BY employee\_id ;

then print out these employees’ ID’s, First names, Last names and their salaries.

Last, do not forget to close the cursor variable.

Hints:

1. Hints for Q1. this is a simple exercise. First run the given codes and get the output.

Then change the LOOP statements, using For dummy\_ind in cursor\_name, ... ,

remember the steps discussed in the class.

1. Hints for Q2. retrieve each employee’s info in department 80, inside the loop, we need if -then, or case to assign the value of the bonus. It is easier to handle two variables, one is bonus for individual employee, another total-bonus, for example:

OPEN c ;

Loop

Fetch cursor\_name into variable\_list ;

EXIT when c%NOTFOUND;

IF comm >= .25 THEN -- Nested IF THEN

IF sal >= 10000 THEN bonus := 1000;

...

END IF; -- end of nested IF

ELSIF (comm < .25 and comm >=.15) THEN -- Nested IF THEN

IF sal >= 10000 THEN bonus := 700;

...

END IF;

ELSE

bonus := 450;

END IF;

Total := total + bonus ;

Bonus := 0; -- reset the bonus to 0 for next employee.

END LOOP;

the correct result should give total bonus: $24,900.

1. Hints for Q3. using cursor with parameter, such as:

Cursor cursor\_name ( owner\_in IN varchar2, table\_in IN varchar2) IS

select column\_name, . . .

from all\_tab\_columns

where owner = owner\_in and table\_name = table\_in;

then later:

OPEN cursor\_name (USER, 'EMPLOYEES') ; -- for basic loop

or

FOR dummy\_indx in cursor\_name (USER, 'DEPARTMENTS') -- for cursor for loop

1. Hints for Q4. Just exercise on cursor for UPDATE, inside the loop, for each individual

employee, when update, needs the clause of where current of cursor\_name:

UPDATE employees

SET salary ...

WHERE current of c;

1. Hints for Q5, cursor variable, similar as example on the notes. Define a strong type, then define a variable for this type. In the loop, open the cursor variable two times.

Recall, for cursor variable, we cannot use Cursor for loop.